Improving Security Techniques in Robot Operating System

Literature Survey

Intro to ROS

- Nodes
- Master
- Publish / Subscribe
- Services
- Graph

Types of attack

- Unauthorized Publishing (Injection)

A node in (plain) ROS may publish data for an arbitrary topic without prior authorization. This may be misused to inject data or commands into an application in order to disturb its operation

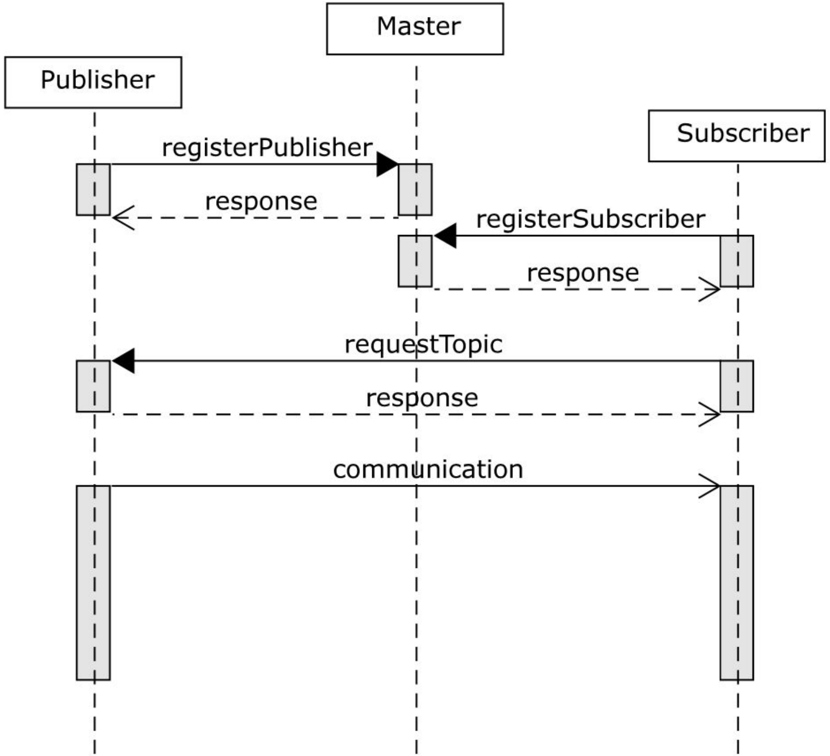
- Unauthorized data access

Every node in ROS may subscribe to every topic within the application. After that, it will receive any data that is published for this topic. This data can contain business-critical information or may be used to reverse-engineer a production process.

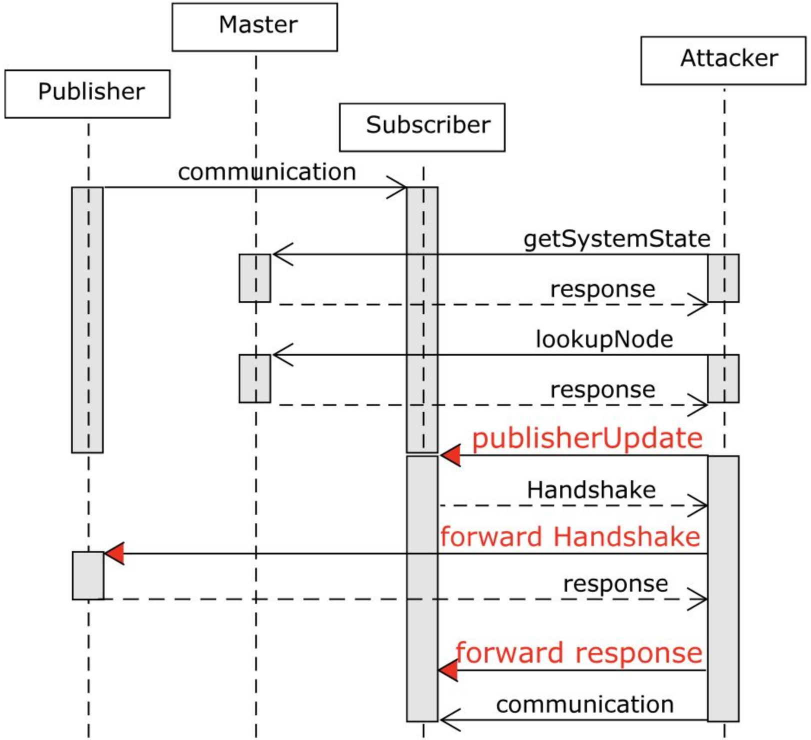
- Denial of Service (DoS) Attack

DoS attacks by publishing a large number of fake data can easily be launched in ROS. The subscriber of this message type will be flooded with bogus messages. This leads to a high processing load on all nodes and potentially to the inability of performing meaningful processing. Since there is no control over which node may publish what data, every node in the network may be used to publish data for a topic which a target node is subscribed to. Later, this can be used for a targeted DoS attack on that node.

Normal Operation of ROS



Attack Operation



Related Work

- changing the network port of the ROS master [1]
  - Issue
- shutting down the ROS master after the application is initialized can make it very hard for an attacker to interfere.
  - However, this also drastically reduces the robustness of an application since a restarting (or late-starting) node cannot join the graph again.

Hardening ROS at application level

- dedicated Authentication Server (AS) which keeps track which ROS node may subscribe to or publish to a topic. In addition, the AS manages the authentication of nodes and generates the topic-specific encryption keys
- Registration of a new ROS node
- After the successful authentication of the new node to the AS, the same procedure is done from the AS to the node to avoid a person-in-the-middle situation.
- a subscriber, the AS sends the subscriber a (digitally signed) list of public signature verification keys related to publishers of the message topics that the subscriber has registered for.

With the presented application-level approach for ROS security, we can already tackle some of the security vulnerabilities. We can prevent unauthorized nodes from publishing and subscribing and thus from injecting false data and from eavesdropping. This is achieved by topic-specific encryption keys which are only handed out to authorized application modules.

only the message content is encrypted, not the message headers. This still allows for frequency analysis of certain message types. Second, the application-level approach cannot prevent nodes from joining the ROS graph or from publishing (even though meaningless) messages to the application which still enables Denial of Service (DoS

Securing the ROS communication channel

- Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)
- Data integrity is secured by using Message Authentication Codes (MACs)
- a subscriber checks if the data of a publisher may be processed and a service checks if it may be consumed by another node
-

Overcoming the problem of plaintext messages

- Ros messages are in plaintext format, which allows easy spoofing and malicious activity
- To prevent this, one way is to implement symmetrical encryption at publisher and subscriber (3DES or AES)
- Safe key sharing mechanism could be through challenge response authentication when a new node is registered with the master
- The subscriber could be asked to reject any messages that in in plaintext.
- To detect if the message is encrypted, entropy of the imcoming data wont work well because messages in ROS are small in size. Other methods such as arithmetic mean, chi-square test etc can be used.